AD-A079 968 FOREIGN TECHNOLOGY DIV WRIGHT-PATTERSON AFB OH F/G 13/6
AUTOMATION INSTITUTE, WARSAW COLLEGE OF ENGINEERING, THE OPERAT--ETC(U)
AUG 79 K SACHA
UNCLASSIFIED FTD-ID(RS)T-0992-79 NL

1 of 1

END
DATE
FILMED
2 80

PHOTOGRAPH THIS SHEET

ADA079968

DTIC ACCESSION NUMBER

LEVEL

INVENTORY

1

FTD-ID(RS)T-0992-79

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR		
NTIS	GRA&I	☒
DTIC	TAB	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION /		
AVAILABILITY CODES		
DIST	AVAIL AND/OR SPECIAL	
A		

DISTRIBUTION STAMP

DDC
RECEIVED
JAN 30 1980
D

DATE ACCESSIONED

79 12 4 011

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DTIC FORM OCT 79 70A

DOCUMENT PROCESSING SHEET

FTD-ID(RS)T-0992-79

FOREIGN TECHNOLOGY DIVISION

AUTOMATION INSTITUTE, WARSAW COLLEGE OF ENGINEERING

THE OPERATING SYSTEM OF THE INTERFACE-MESSAGE COMPUTER SYSTEM FOR TRAFFIC CONTROL ON THE CENTRAL RAILROAD TRUNKLINE

by

Krzysztof Sacha

EDITED TRANSLATION

FTD-ID(RS)T-0992-79 15 August 1979

MICROFICHE NR: FTD-79-C-001105

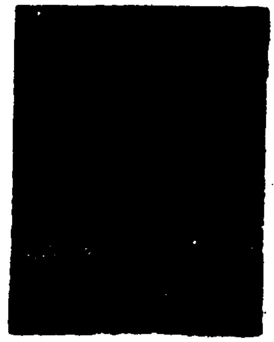
AUTOMATION INSTITUTE, WARSAW COLLEGE OF ENGINEERING
THE OPERATING SYSTEM OF THE INTERFACE-MESSAGE COMPUTER SYSTEM FOR TRAFFIC CONTROL ON THE CENTRAL RAILROAD TRUNKLINE

By: Krzysztof Sacha

English pages: 13

Source: Informatyka, Vol. 13, Nr. 3, 1978,
pp. 5-8

Country of origin: Poland
Translated by: SCITRAN
F33657-78-D-0619
Requester: RCA
Approved for public release; distribution unlimited.



Dr. of Engineering KRZYSZTOF SACHA completed his studies with distinction in the Electronics Department of Warsaw College of Engineering in 1973 and defended his doctoral dissertation in computer theory in 1976. He is now working on software problems at the Automation Institute of Warsaw College of Engineering.

KRZYSZTOF SACHA

Automation Institute, Warsaw College of Engineering

The Operating System of the Interface-Message Computer System for Traffic Control on the Central Railroad Trunkline

As a result of the country's economic development, the existing railroad network has not been able to keep pace with growing needs. Hence construction was begun on a new traffic route linking the Silesian industrial district with the center of the country. The Central Railroad trunkline will meet all modern requirements. Trains will run at speeds exceeding 200 km/h, and during periods of heaviest traffic the time interval between successive trains will be no more than several minutes. Under such conditions, only a computer system for controlling train traffic can insure effective and safe trunkline utilization. The system should maintain full control over the state of all track equipment (e.g., switches) and over the current speeds of all trains. The design of such an automatic control system was developed by the Automation Institute of the Warsaw College of Engineering (3). According to the present schedule, implementation of the control system will begin in the early 80's.

The Traffic Control System on the Central Railroad Trunkline

The design establishes a two-layered Control Center, in the form of a satellite network, shown on Fig. 1: a main computer of average power output and a system of small interface-message computers. Implementation of the system is anticipated to be in two stages. In the first, the basic decisions are made by the dispatcher assisted by a traffic tracing system, based on domestic minicomputers of the MERA-300 series. In the second--the dispatcher's function is assumed by the main computer, for which the tracing computers will become interface-message computers. During transition from the first to the second stage, programming of the interface-message computers will not change very much. The entire operating system, particularly, will be transferred without any changes.

In the first stage, train traffic from Warsaw to Katowice, about 300 km, is controlled centrally by three dispatchers, each of whom oversees a route of about 100 km. The dispatchers work in the Control Center, using an illuminated track diagram showing the actual situation on the trunkline, and keyboards for giving commands. Tracing system tasks include the following:

--constant control of the condition of all track equipment (switches, semaphores, track sections)

--tracing of train switching on the trunkline and illustrating actual positions on the illuminated diagram

--recording and monitoring the execution of all of the dispatcher's commands (such as, reset switch, set train speed).

It is essential that the interface-message computers operate in real-time mode, with average time interval between two successive reports on the conditions of the trunkline on the order of several milliseconds.

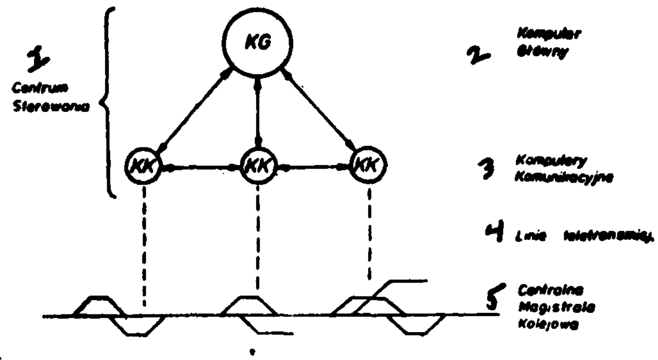


Fig. 1. Control System Structure

Key:

1. Control center
2. Main computer
3. Interface-message computer
4. Television signal transmission lines
5. Central railroad trunkline

One MERA-300 series minicomputer, without external storage, can oversee about 100 km of trunkline. Hence, three machines, interconnected, are indispensable to the tracing system. For communicating with the dispatcher, each interface-message computer is equipped with a printer, on which all communication and operating documentation is printed, and the dispatcher's keyboard. Connected to the interface-message computers are receivers of the BUSZ television signal transmission system which is the intermediary in passing information between the Control Center and the basic layer equipment, installed on the Central Railroad Trunkline.

The key element in programming the interface-message computers is the operating system which controls all the processes of collecting and processing information flowing in from the track equipment. The basic organizational solutions of the operating system are described further in this article.

Functions of the RTX-S Operations System

RTX-S is a multi-program (multi-task) real-time operations system, organizing the synchronous work of many independent programs (tasks) and planning their execution in a time function and in functions occurring in the events system. Each task is an independent program realizing an assigned portion of the activities of the tracing system. Furthermore, RTX-S insures, within a limited range, the control of communication with external equipment and implements certain service functions. All programs comprising the programming of the interface-message computer are executed under the supervision of this system.

In particular, the functions of the RTX-S system include:

1) accepting interruptions, i.e., decoding causes of interruptions, actuating the subroutine for servicing the interruption, and returning to the execution of the interrupted program (task)

2) scheduling tasks according to their individual priorities

3) counting time and cyclical or one-time activation of tasks at specified moments

4) servicing external events, dependent on activation of tasks related to random interruptions

5) implementing extracode instructions expanding the list of commands to MERA by indirect addressing and a multi-level subroutine jump storing the tracing in the stack

6) control of communication with nonstandard external equipment (with teletransmission system receivers, with keyboards and other interface-message computers)

7) loading binary tapes to operating storage with readout correctness check.

Communication of interface-message computer with dispatcher is via special tasks which are not part of the operating system.

Organizationally, the RTX-S system is made up of a series of modules which perform particular command and servicing functions. The basic structure of the system, which comprises an integral whole, is made up of the modules executing functions 1 to 5. The remaining modules are included in the system structure as options.

Principles of Operation of the RTX-S Operating System

Processing programs, operating under RTX-S system control, can be executed on one of two levels. The lower level is comprised of interruption servicing subroutines, executed in an inoperative interruption system, and insuring very rapid interface-message computer reaction to interruption-signalled events. The upper level is comprised of synchronously executed, in an operative interruption system, tasks which are the basic organization form of processing programs. All tasks (and only tasks) can make use of extracode instructions.

Up to 30 tasks can be executed synchronously, scheduled according to established priorities. At any moment the tasks may be in one of four states:
M (dead), G (ready), W (executed), Z (delayed). Fig. 2 shows the states of tasks. Transitions between states occur for the following reasons:

1) start of task (M→G) occurs as a result of:

--execution of START command by another processing program

--notice of interruption related to a given task

--time lapse, after cyclical task should be activated

2) delay of task (W→Z) occurs as a result of the execution of the task by the DELAY command, delaying the task for the time given as the command parameter

3) restart of task (Z→G) occurs after a time lapse specified in the previously executed DELAY command

4) end of task (W→M) occurs after execution of the EXIT command.

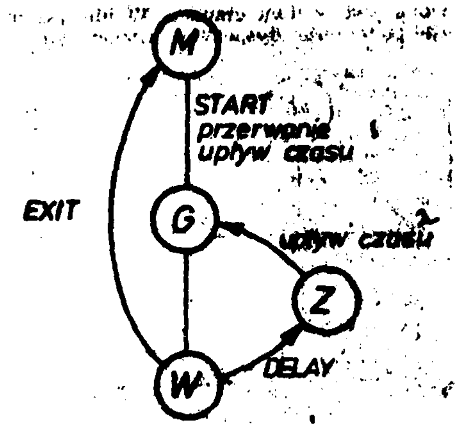


Fig. 2. State of tasks

Key:
1. Time lapse interruption
2. Time lapse

Execution of the task by the DELAY or EXIT command removes the task from the W state and interrupts its further realization. At this moment, control is assumed by the scheduling program which selects the ready program (state G) with the highest priority and transfers it to state W, causing it to be put into being. And so, scheduling is done without dispossession, at constant task priorities.

The basic information for implementing the above-described mechanisms for controlling tasks is stored in task state boards, containing full descriptions of all tasks. The address of task states in storage is the identifier of the task in the system.

The description of the task covers:

--the state of the data processor registers at the moment of the delay of the task (PSW)

--the actual state of the task

--the period of repetition (cyclical tasks) and time to nearest activation

--stack of tracings of multi-level (extracode) subordinate jump.

The general organizational concept of the RTX-S operating system in large measure uses the mechanism described in (2), but the main emphasis is on maximizing the speed and effectiveness of action. Figure 3 shows the basic elements of the system's internal structure

The action of particular modules of the system consists of converting the information in the system boards. In addition to the task state boards, the following are used:

--an interruption servicing subroutine board, assigning servicing subroutine addresses to interruption numbers

--an extracode board, assigning addresses for the required subroutine functions to extracode codes

--an external events code, assigning identifiers of activated tasks to interruption numbers

--a task board containing a list of identifiers for all tasks present in the task system.

This solution will make it possible to easily change the system configuration by exchanging the content of all the appropriate boards without having to modify any programming elements.

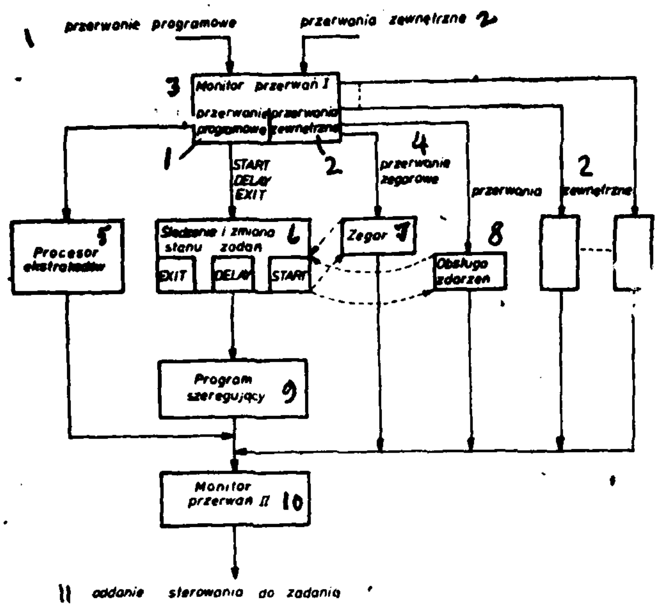


Fig. 3. Block diagram of RTX-3 operating system

Key:

1. Program interruptions
2. External interruptions
3. Interruption monitor I
4. Clock interruptions
5. Extracode processor
6. Tracing and changing tasks
7. Clock
8. Events servicing
9. Scheduling program
10. Interruption monitor II
11. Release of control to task

An example of the expansion of the basic structure of the system is the addition of a multi-access communication subroutine between two interface-message computers. The subroutine is comprised of three modules, one of which is executed directly after generating the subroutine, and the two remaining are executed later, during servicing of the reported interruptions.

Due to the importance of the intercomputer connection, the rule of return transfer of each character was accepted. In case of error detection, the entire transmission is repeated. If after three repeats an error-free connection cannot be obtained, an alarm is sounded. Tasks can make use of the subroutine by generating extracodes. The subroutine can be included in the system by entering the starting addresses of all modules in the proper fields of the extracode board and the interruption servicing subroutine board.

Synchronization of Tasks

The first group of synchronization problems are those resulting from mutual conditioning of tasks. Acceptance of the nondispossessing scheduling rule insured mutual exclusion of tasks during access to common data. Only matters of information transfer between cooperating tasks in the producer-user system and problems of insuring the required sequence of task execution on time, remained to be solved. The first of these problems was solved by implementing the data buffer, serviced by the loading and reports-receiving operations from the buffer, whick make up the send and receive operations described in (1). This is based on the ability of several producers and users to use the buffer, each of whom must receive the information produced and placed in the buffer. The second RTX-S system problem is solved by the clock synchronization mechanism (time) and by the START command, allowing the task being executed to activate any other task.

All system commands used to communicate tasks among each other and tasks from the operating system are listed in Table 1. These commands are issued extracode.

Time conditions of tasks are realized in the RTX-S system by a self-contained module, shown in Fig. 3 as "clock". This module is activated each time after receiving a time lapse interruption reported by a highly stable time generator. From the processor standpoint the clock module supplies two kinds of synchronization tools. The first is the DELAY command described in Table 1, for which the clock module counts down the time for delaying the task. The second improvement is the ability to declare a task to be cyclical, which causes its automatic activation in a specified repeat period. Accuracy of count-down time was assumed in the tracing system to be 100 ms.

Table 1. System Commands

Command	Designation in Macro-assembly	Content	Parameters
START (I)	SA*	Activate task	Identifier I of the activated task
DELAY (T)	NN*	Delay executed task for time T	T period of task delay
EXIT	SD*	Complete task	
LOAD (A)	AL*	Load report from A to buffer	Address A of report
PULL (A, K)	AP*	Pull report from buffer	Address A of report transfer and K user identification

The last type of task synchronization, implemented in the RTX-S system, is planning execution of tasks related to external events, signalled by

interruptions of particular numbers. These functions are performed by the system's external events servicing module, activating related tasks at the moment of interruption report.

Table 2 lists table synchronization methods performed by the RTX-S system.

The RTX-S system me-`anisms described so far are universal methods, insuring a widely understood synchronization of arbitrary synchronous tasks. A different approach requires that tasks and interruption servicing subroutines be coordinated and executed immediately upon report of interruption, without waiting for task completion. The solution was based on a set of specialized service commands realized by the operating system during interruption shutoff. This automatically solves the problem of mutual exclusion of processes and is also very time-saving. In view of the modular structure of the RTX-S system and the orientation of its operation to converting boards, the addition of new commands to the existing configuration is not difficult. The list of commands accessible both to tasks and to interruption servicing subroutines covers, among others, the START command, commands for communicating with nonstandard external equipment and data buffer servicing commands.

Table 2. Methods for Synchronizing Tasks

Type of conditioning	Synchronization method
Mutual conditioning of tasks	--nondispossessing scheduling --data buffer --START, EXIT, DELAY commands
Time conditioning	--repeated activation of cyclical tasks --DELAY command
External conditioning	--activation of tasks related to specified interruptions

Concluding Remarks

The fact that nondispossessing scheduling occurs in the RTX-S system without any safeguards against garbling the tasks being executed, may be cause for some concern. The solution, however, is permissible in a specialized system in which only debugged and pretested programs are executed. On the other hand, the solution has two important advantages: First, it is time-saving because it restricts the intervention of the operating system to a minimum. Second, it automatically insures mutual exclusion of tasks, which means that the time- and storage-consuming wait and signal (1) semaphore operations need not be used.

Laboratory studies of tracing system programming showed that the most critical parameter of the system is storage capacity. Therefore, the RTX-S, as a specialized system, is equipped with only those mechanisms which are indispensable to the described application. The function of communication with the operator, useful in the program debugging phase operating under the supervision of the RTX-S system, is not being used.

A less critical limitation is the time for execution of tasks. At the assumptions cited above, the load on the system averaged over a 1-second period did not exceed 70%, even when external events were unfavorable. A large decrease in speed is due to lack of indirect addressing equipment in MERA-300 series machines. Even economical use of extracode instructions with indirect addressing causes a 6- to 8-fold extension of time for program execution. The speed of the RTX-S system is described by the operations execution times shown in Table 3.

Table 3. Approximate Times of RTX-S System Operation

System Function	Execution Time
Indirect addressing instructions	~300 microsec.
Interruption servicing: --beginning of subroutine --return	 ~110 microsec. ~ 60 microsec.
START, EXIT commands	~250 microsec.

REFERENCES

[1] Hansen P. B.: On the operating system principles. Prentice Hall, inc., Englewood Cliffs, New Jersey 1973

[2] Misiurewicz P.: Simple real-time executive for small minicomputer. Referat na konferencji „Soccoo". Tallin, maj 1976

[3] Traczyk W., Misiurewicz P., Kostro J., Perkowski M., Rydzewski A.: System sterowania ruchem na centralnej magistrali kolejowej. Materiały przygotowane z okazji 25-lecia wydziału elektroniki PW. Warszawa, 25—27 lutego 1976.

DISTRIBUTION LIST

DISTRIBUTION DIRECT TO RECIPIENT

ORGANIZATION	MICROFICHE
A205 DMATC	1
A210 DMAAC	2
B344 DIA/RDS-3C	9
C043 USAMIIA	1
C509 BALLISTIC RES LABS	1
C510 AIR MOBILITY R&D LAB/FIO	1
C513 PICATINNY ARSENAL	1
C535 AVIATION SYS COMD	1
C591 FSTC	5
C619 MIA REDSTONE	1
D008 NISC	1
H300 USAICE (USAREUR)	1
P005 DOE	1
P050 CIA/CRB/ADD/SD	2
NAVORDSTA (50L)	1
NASA/NST-44	1
AFIT/LD	1
LLL/Code L-389	1
NSA/1213/TDL	2

ORGANIZATION	MICROFICHE
E053 AF/INAKA	1
E017 AF/RDXTR-W	1
E403 AFSC/INA	1
E404 AEDC	1
E408 AFWL	1
E410 ADTC	1
FTD	
CCN	1
ASD/FTD/NIIS	3
NIA/PHS	1
NIIS	2